

BAB II

TINJAUAN PUSTAKA

Pada bab ini berisi landasan teori yang berhubungan dengan penelitian. Tinjauan pustaka meliputi : Android, Google Maps API, Logika Fuzzy, Fuzzy Tahani, Objek Wisata, Basis Data, dan SQLite.

2.1 Android

2.1.1 Pengenalan Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi [3]. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android tidak terkait ke satu vendor *smartphone*, beberapa *smartphone* berbasis android yaitu HT, Motorola, Samsung, LG, Asus, dan lain-lain. Tidak hanya menjadi sistem dalam *smartphone* tapi juga dalam sistem *tablet PC*.

Android merupakan *platform* yang lengkap, terbuka dan bebas yang artinya [3] :

- a. Lengkap artinya para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* android. Sistem operasinya aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan peluang untuk pengembangan aplikasi.
- b. Terbuka artinya *platform* android disediakan melalui lisensi terbuka (*open source*) sehingga pengembang dapat dengan bebas mengembangkan aplikasi.
- c. Bebas artinya tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Aplikasi android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

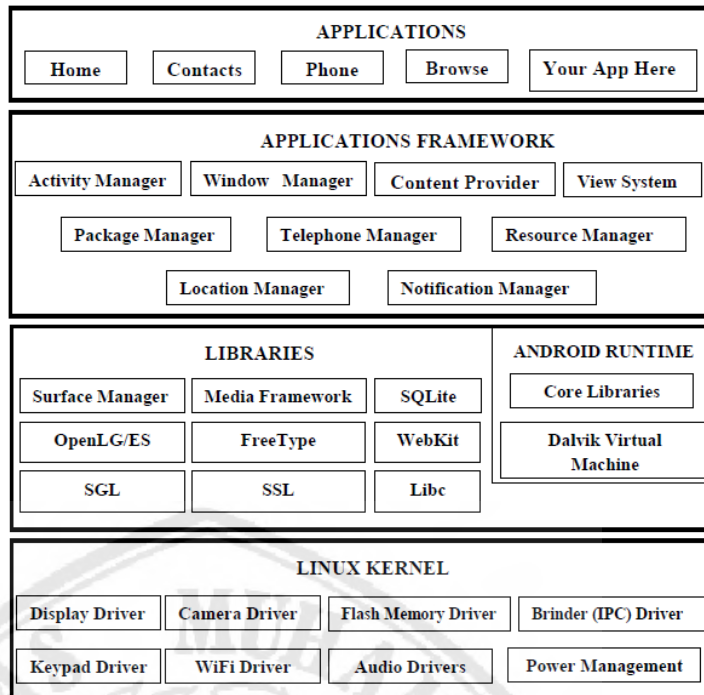
2.1.2 Fitur-fitur Android

Android tersedia secara terbuka bagi manufaktur perangkat keras untuk memodifikasi sesuai kebutuhan. Konfigurasi perangkat android tidak sama antara satu perangkat dengan perangkat lainnya, namun android sendiri mendukung fitur-fitur tertentu, menurut [3] fitur-fitur android sebagai berikut :

- a. Penyimpanan (*storage*) menggunakan SQLite yang merupakan database relasional yang ringan untuk menyimpan data.
- b. Koneksi (*connectivity*) mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (termasuk A2DP dan AVRCP), WiFi, LTE dan WiMAX.
- c. Pesan (*messaging*) mendukung SMS dan MMS.
- d. Web browser menggunakan *open source* WebKit termasuk di dalamnya *engine* Chrome V8 Java Script.
- e. Media yang didukung antara lain: H.263, H.264 (3GP atau MP4), MPEG-4 SP, AMR, AMR-WB (3GP container). ACC, HE-ACC (MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF dan BMP.
- f. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- g. *Hardware* terdapat *accelerometer* sensor, kamera, kompas digital, *proximity* sensor dan GPS.
- h. Multi-touch.
- i. Multi-tasking.
- j. Dukungan flash.

2.1.3 Arsitektur Android

Arsitektur android dapat dijelaskan dan digambarkan pada gambar berikut [5] :



Gambar 2.1 Arsitektur Android

Penjelasan mengenai gambar di atas sistem operasi android terbagi menjadi :

- Applications* : layer dimana akan berhubungan dengan aplikasi saja. Biasanya aplikasi yang diunduh dan diinstal dan semua aplikasi yang dibuat terletak pada tingkat *applications* seperti : *phone, contact, browser*.
- Applications framework* : semacam built-in yang tertanam dalam sistem operasi android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.
- Libraries* : semua kode program yang menyediakan layanan-layanan utama sistem operasi android. Contoh *library* SQLite yang menyediakan dukungan basis data sehingga aplikasi android dapat untuk menyimpan data.
- Android Runtime* : kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi android. Layer yang membuat aplikasi android dapat dijalankan prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin

yang membentuk dasar kerangka aplikasi android. *Runtime* android terbagi menjadi dua yaitu :

1. *Core LibrariesI* : aplikasi android dibangun dalam bahasa java, sementara dalvik sebagai virtual mesinnya bukan virtual mesin java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java yang ditangani oleh core libraries.
 2. *Dalvik Virtual Machine* : virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangannya yang mampu membuat *linux kernel* untuk melakukan threading dan manajemen tingkat rendah
- b. *Linux Kernel* : layer inti sistem operasi android itu berada. Berisi berkas-berkas sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers* dan sistem-sistem android lainnya.

2.2 Google Maps API

2.2.1 Menenal Google Maps

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer untuk pemetaan digital. Aplikasi ini diperkenalkan pada Februari 2005 dan merupakan evolusi penyajian peta dalam bentuk digital [9]. Saat ini Google Maps untuk perangkat android telah mencapai Google Maps v3, yang tentu saja berbeda dengan sebelumnya yaitu Google Maps v1 dan Google Maps v2. Google Maps v3 ini diluncurkan pada Maret 2013 dan menawarkan lebih banyak fungsionalitas dari pada versi sebelumnya seperti pembuatan peta 3D.

Google Maps menyediakan layanan berupa pemetaan jalan, rute, dan navigasi untuk berbagai rute perjalanan sehingga dapat digunakan untuk mempercepat pencarian sebuah lokasi dalam waktu yang singkat, dan juga menunjukkan jalan mana saja yang harus ditempuh untuk mencapai tujuan. Teknologi Google Maps menggunakan citra satelit untuk melakukan pemetaan objek yang ada di permukaan bumi secara *realtime*,

dalam hal ini peta yang ada pada Google Maps di update dalam kurun waktu tertentu.

Google Maps dapat ditambahkan dalam web, blog maupun aplikasi *mobile* dengan menggunakan Google Maps API. Dalam penerapannya Google Maps juga dapat disesuaikan dengan kebutuhan aplikasi yang akan dibuat [9].

2.2.2 Google Maps API V2

Google telah menyediakan layanan bagi para developer aplikasi android untuk menggunakan API Google Maps pada aplikasi mereka. API sendiri adalah singkatan dari *Application Programming Interface*. Selain itu, dokumentasi dan forum yang membahas penggunaan API Google Maps pada aplikasi android juga sudah tersedia banyak di situs developer android.

Perkembangan penggunaan Google Maps di android yang ada saat ini dimulai dengan adanya Google Maps V1 yang penggunaannya telah dihentikan pada akhir tahun 2012. Mulai tahun 2013, aplikasi android yang ingin menampilkan Google Maps harus menggunakan layanan Google Maps V2. Ada perbedaan yang mencolok dalam penerapan *source code* antara keduanya. Mulai dari penggunaan SHA1 yang menggantikan MD5 untuk mendapatkan Google API Key hingga penggunaan *Fragment* yang menggantikan *MapView*. Kita juga harus menginstall library *google-play-service* terlebih dahulu pada android SDK [2].

Sebagaimana pada pengembangan Google Maps V1, pada pengembangan aplikasi yang menggunakan Google Map V2, kita juga mendapat akses untuk menambahkan *marker* (tanda lokasi), baik dengan ikon yang telah tersedia maupun menggunakan ikon kita sendiri. *Marker* juga dapat kita lengkapi dengan keterangan, sehingga ketika ditekan dan muncul informasi yang disisipkan.

Pendeteksi lokasi pengguna juga dapat kita tambahkan pada pengembangan aplikasi yang menggunakan Google Maps, baik menggunakan GPS, jaringan seluler, maupun jaringan WiFi.

2.3 Logika Fuzzy

Logika *fuzzy* merupakan salah satu komponen pembentuk *soft computing*. Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran logika *fuzzy* tersebut.

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama[4].

2.3.1 Himpunan Fuzzy

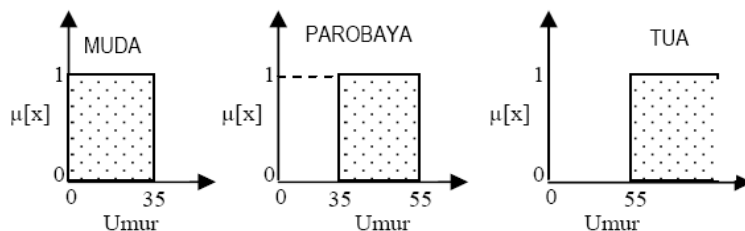
Himpunan *fuzzy* adalah pengelompokan sesuatu berdasarkan variabel bahasa (*linguistik variable*), yang dinyatakan dengan fungsi keanggotaan, dalam semesta U . Keanggotaan suatu nilai pada himpunan dinyatakan dengan derajat keanggotaan yang nilainya antara 0.0 sampai 1.0.

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan [1], yaitu :

1. Satu(1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Contoh, misalkan variabel umur dibagi menjadi 3 kategori, yaitu :

MUDA	umur < 35 tahun
PAROBAYA	$35 \leq \text{umur} \leq 55$ tahun
TUA	umur > 55 tahun



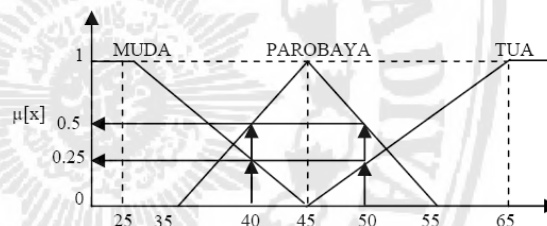
Gambar 2.2 Himpunan : MUDA, PAROBAYA, dan TUA

Pada gambar diatas dapat dijelaskan bahwa :

1. Apabila seseorang berusia 34 tahun, maka ia dikatakan MUDA ($\mu_{MUDA}(34) = 1$);
2. Apabila seseorang berusia 35 tahun, maka ia dikatakan TIDAK MUDA ($\mu_{MUDA}(35) = 0$);
3. Apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK MUDA ($\mu_{MUDA}(34 \text{ th} - 1 \text{ hari}) = 0$);

Dari contoh diatas bisa dikatakan bahwa pemakaian himpunan crisp untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan.

Himpunan *fuzzy* digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam 2 himpunan berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dsb. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaanya.



Gambar 2.3 Himpunan *fuzzy* untuk variabel umur

Pada gambar diatas dapat dilihat bahwa :

1. Seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan $\mu_{MUDA}(40) = 0,25$; namun dia juga termasuk dalam himpunan PAROBAYA dengan $\mu_{PAROBAYA}(34) = 0,5$.
2. Seseorang yang berumur 50 tahun, termasuk dalam himpunan MUDA dengan $\mu_{TUA}(50) = 0,25$; namun dia juga termasuk dalam himpunan PAROBAYA dengan $\mu_{PAROBAYA}(50) = 0,5$.

Pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 dan 1. Pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A(x) = 0$ berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai

keanggotaan *fuzzy* $\mu_A(x) = 1$ berarti x menjadi anggota penuh pada himpunan A .

Himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, dan TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu :

1. Variabel *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh : umur, temperatur, permintaan, dsb.
2. Himpunan *fuzzy*, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.
3. Semesta pembicaraan, adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif.
4. Domain himpunan *fuzzy*, adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

2.3.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering disebut dengan derajat keanggotaan) yang memiliki interval antara 0 hingga 1. Salah satu cara yang dapat digunakan

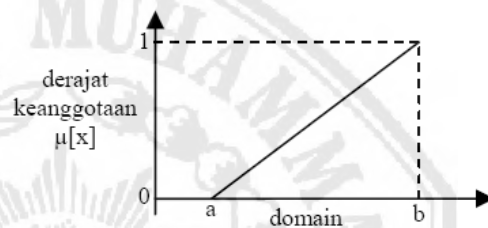
untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan, yaitu :

1. Representasi Linier

Pada representasi linier, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati satu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* linier, yaitu :

a. Representasi Linier Naik

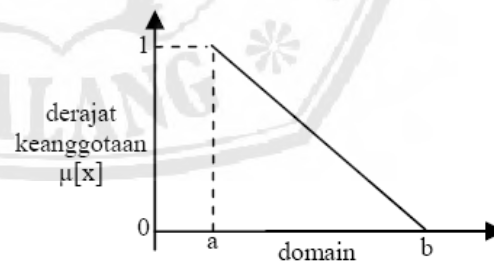
Kenaikan himpunan dimulai pada dominan yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai dominan yang memiliki derajat keanggotaan lebih tinggi.



Gambar 2.4 Representasi Linier Naik

2. Representasi Linier Turun

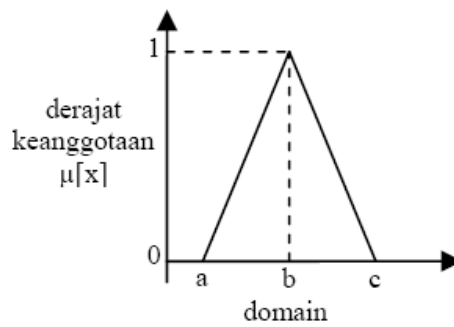
Garis lurus dimulai dari nilai dominan dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai dominan yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.5 Representasi Linier Turun

3. Representasi Kurva Segitiga

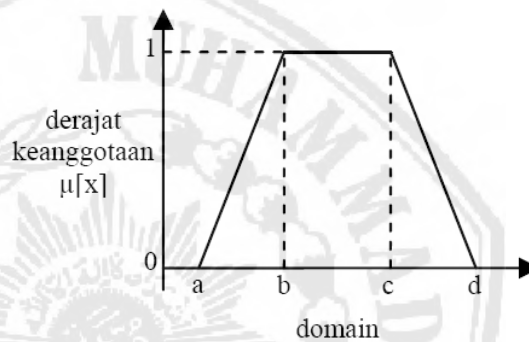
Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier).



Gambar 2.6 Representasi Kurva Segitiga

4. Representasi Kurva Trapesium

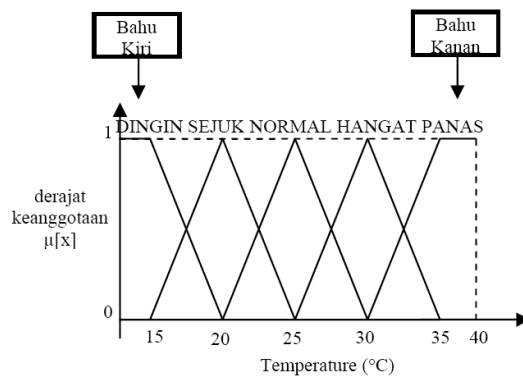
Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.7 Representasi Kurva Trapesium

5. Representasi Kurva Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap berada pada kondisi PANAS. Himpunan *fuzzy* 'bahu', bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.



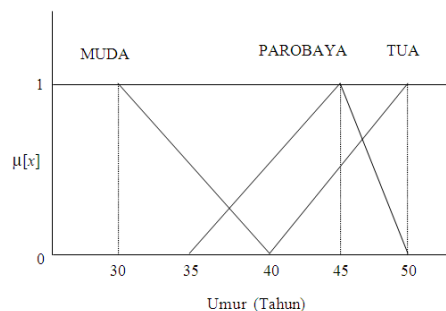
Gambar 2.8 Representasi Kurva Bahu

2.3.3 Fuzzy Tahani

Basisdata yang umumnya kita gunakan, memiliki data yang lengkap dalam setiap tabelnya. Demikian pula, apabila hendak dibuat suatu *query*, maka *query* itupun harus menggunakan data yang ada pada tabel dan kata-kata kunci yang berlaku di SQL. Apabila kita memiliki data yang kurang lengkap, mengandung ketidakpastian dan ambigu, maka penggunaan basisdata biasa menjadi sulit untuk dilakukan. dari sinilah logika *fuzzy* dapat dimanfaatkan untuk mengantisipasi pemanipulasian data dalam basisdata yang mengandung ketidakpastian, baik dari sisi data maupun *query*-nya [1].

Selama ini suda ada beberapa penelitian tentang basisdata *fuzzy*. Salah satu diantaranya adalah model Tahani. Basisdata fuzzy model Tahani masih tetap menggunakan relasi standar, hanya saja model ini menggunakan teori himpunan *fuzzy* untuk mendapatkan informasi pada *query*-nya.

Misalkan dalam mengkategorikan usia karyawan ke dalam himpunan : MUDA, PAROBAYA, dan TUA.



Gambar 2.9 Fungsi Keanggotaan Untuk Variabel Usia

Fungsi keanggotaan :

$$\mu_{MUDA}[x] = \begin{cases} 1; & x \leq 30 \\ \frac{40-x}{10}; & 30 \leq x \leq 40 \\ 0; & x \geq 40 \end{cases}$$

$$\mu_{PAROBAYA}[x] = \begin{cases} 0; & x \leq 35 \text{ atau } x \geq 50 \\ \frac{x-35}{10}; & 35 \leq x \leq 45 \\ \frac{50-x}{5} & 45 \leq x \leq 50 \end{cases}$$

$$\mu_{TUA}[x] = \begin{cases} 0; & x \leq 40 \\ \frac{x-40}{10}; & 40 \leq x \leq 50 \\ 1; & x \geq 50 \end{cases}$$

Pada tabel berikut menunjukkan tabel karyawan berdasarkan umur dengan derajat keanggotaannya pada setiap himpunan.

Tabel 2.1 Karyawan Berdasarkan Umur

NIP	Nama	Umur	Derajat Keanggotaan [x]		
			MUDA	PAROBAYA	TUA
01	Lia	30	1	0	0
02	Iwan	48	0	0,4	0,8
03	Sari	36	0,4	0,1	0
04	Andi	37	0,3	0,2	0
05	Budi	42	0	0,7	0,2

Rentang nilai yang disarankan antara 0 hingga 1, dengan rekomendasi tertinggi adalah 1 dan berangsur tidak disarankan apabila nilai mendekati 0.

Ada beberapa *query* yang bisa diberikan, misalkan : siapa sajakah karyawan paling muda ?

SELECT NAMA FROM KARYAWAN WHERE Umur="MUDA"

2.3.4 Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan

sering dikenal dengan nama *fire strength*. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu :

1. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. *Fire strength* sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antarelemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

2. Operator OR

Operasi ini berhubungan dengan operasi union pada himpunan. *Fire strength* sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antarelemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. *Fire strength* sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A(x)$$

2.4 Objek Wisata

2.4.1 Pengertian Objek Wisata

Dalam dunia kepariwisataan, segala sesuatu yang menarik dan bernilai untuk dikunjungi dan dilihat, disebut atraksi atau lazim pula dinamakan objek wisata. Objek wisata adalah segala sesuatu yang mempunyai daya tarik, keunikan dan nilai yang tinggi, yang menjadi tujuan wisatawan datang ke suatu daerah tertentu.

2.4.2 Syarat-Syarat Objek Wisata

Sebuah objek wisata yang baik harus dapat mendatangkan wisatawan sebanyak banyaknya, menahan mereka ditempat objek wisata dalam waktu yang cukup lama dan memberi kepuasan kepada wisatawan

yang datang berkunjung. Untuk mencapai hasil itu, beberapa syarat harus dipenuhi, yaitu :

1. Kegiatan (*act*) dan objek (*artifact*) yang merupakan objek wisata itu sendiri harus dalam keadaan yang baik.
2. Karena objek wisata itu disajikan dihadapan wisatawan, maka cara penyajiannya harus tepat.
3. Objek wisata adalah terminal dari suatu mobilitas spasial atau perjalanan. Oleh karena itu juga harus memenuhi semua determinan mobilitas spasial, yaitu akomodasi, transportasi dan promosi serta pemasaran.
4. Keadaan di objek wisata harus dapat menahan wisatawan cukup lama.
5. Kesan yang diperoleh wisatawan waktu menyaksikan atraksi wisata harus diusahakan supaya bertahan selama mungkin.

2.4.3 Karakteristik Objek Wisata

Ada 3 karakteristik utama dari objek wisata yang harus diperhatikan dalam upaya pengembangan suatu objek wisata tertentu agar dapat menarik dan dikunjungi banyak wisatawan. Karakteristik tersebut antara lain :

1. Daerah itu harus mempunyai apa yang disebut sebagai *something to see* yang berarti tempat tersebut harus ada objek wisata dan atraksi wisata yang berbeda dengan apa yang dimiliki oleh daerah lain.
2. Daerah tersebut harus tersedia apa yang disebut dengan *seomething to do* yang berarti tempat tersebut selain banyak yang dapat disaksikan, harus disediakan pula fasilitas reaksi yang dapat membuat wisatawan betah tinggal lebih lama di tempat itu.
3. Daerah tersebut harus tersedia apa yang disebut dengan *something to buy* yang berarti tempat tersebut harus ada fasilitas untuk berbelanja, terutama barang-barang souvenir dan kerajinan tangan rakyat sebagai oleh-oleh untuk dibawa pulang.

2.5 Basis Data

2.5.1 Pengertian Basis Data

Pengertian basis data adalah sebagai berikut [8] :

1. Menurut Stephens dan Plew, adalah mekanisme yang digunakan untuk menyimpan informasi atau data.
2. Menurut Silberschatz, dkk mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.
3. Menurut Ramakrishnan dan Gehrke menyatakan basisdata sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.
4. Menurut McLeod, dkk adalah kumpulan seluruh sumber data berbasis komputer milik organisasi.

Jadi, basisdata adalah suatu koleksi data yang saling berhubungan secara logis dan menggambarkan integrasi antara suatu tabel dengan tabel lainnya, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi [8].

Berikut ini merupakan alasan dari penggunaan basisdata :

1. Padat. Tidak perlu lagi membuat arsip kertas dalam ukuran besar.
2. Kecepatan. Mesin dapat mendapatkan kembali dan mengubah data jauh lebih cepat daripada yang manusia dapat lakukan.
3. Mengurangi pekerjaan yang membosankan. Rasa bosan dari proses memelihara arsip-arsip berupa kertas dapat dikurangi.
4. Aktual. Informasi yang terbaru dan akurat selalu tersedia disetiap waktu ketika dibutuhkan.

Terdapat delapan keuntungan dengan menggunakan pendekatan basisdata, yaitu :

1. Redundansi dapat dikurangi.
2. Ketidak konsistenan dapat dihindari.
3. Data dapat dibagikan.
4. Standar-standar dapat diselenggarakan.
5. Pembatasan keamanan dapat diselenggarakan.
6. Integritas dapat dipertahankan.

7. Keperluan yang bertentangan dapat diseimbangkan.
8. Tersedianya dukungan untuk atraksi.

Maksud dari *integrated* adalah basisdata merupakan penggabungan beberapa berkas data yang berbeda, dengan mambatasi pengulangan baik keseluruhan file ataupun sebagian. Pengertian *shared* artinya adalah data individu dalam basisdata dapat digunakan secara bersamaan antara beberapa pengguna yang berbeda.

2.5.2 Penjelasan SQLite

SQLite merupakan sebuah sistem manajemen basis data yang simpel dalam penggunaannya dan dapat disisipkan dalam aplikasi android. Apabila kita ingin mengakses asis data di web server, kita perlu koneksi internet, tetapi pada SQLite kita tidak perlu tersambung dengan internet. kita dapat menambah, mengedit, dan menghapus data di dalamnya, serta dapat menjalankan sebuah sintaks SQL [2]. SQLite memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. Tidak seperti pada paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency*, *times*, dan secara keseluruhan lebih sederhana. seluruh elemen basis data (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

2.5.3 Fitur SQLite

SQLite mengimplementasikan hampir seluruh elemen standar, termasuk transaksi yang bersifat *atomic*, *consistency database*, *isolated*, *and durable*, trigger dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk

sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basis data, utamanya ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan [6]. Beberapa proses ataupun thread dapat berjalan secara bersamaan dan mengakses basis data yang sama tanpa mengalami masalah. Hal ini disebabkan karena akses baca data dilakukan secara paralel. Sementara itu akses tulis lain data hanya bisa dilakukan; jika tidak, proses tulis tersebut akan gagal dan mengembalikan kode kesalahan. Sebuah program yang mandiri dinamakan SQLite disediakan dan bisa digunakan untuk mengeksekusi kueri dan manajemen file-file basis data SQLite.

